# Evaluation of cross-validation strategies in sequence-based binding prediction using Deep Learning

Angela Lopez-del Rio,[*,†,‡,¶,§] Alfons Nonell-Canals,[‡] David Vidal,[‡] and Alexandre Perera-Lluna[†,¶,§]

†*B2SLab, Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial, Universitat Politècnica de Catalunya, 08028 Barcelona, Spain.*

‡*Mind the Byte S.L., 08007 Barcelona, Spain.*

¶*Networking Biomedical Research Centre in the subject area of Bioengineering, Biomaterials and Nanomedicine (CIBER-BBN) 28029, Madrid, Spain*

§*Department of Biomedical Engineering, Institut de Recerca Pediàtrica Hospital Sant Joan de Déu, Esplugues de Llobregat, 08950 Barcelona, Spain.*

E-mail: angela@mindthebyte.com

**Abstract**

Binding prediction between targets and drug-like compounds through Deep Neural Networks have generated promising results in recent years, outperforming traditional machine learning-based methods. However, the generalization capability of these classification models is still an issue to be addressed. In this work, we explored how different cross-validation strategies applied to data from different molecular databases affect to the performance of binding prediction proteochemometrics models. These strategies are: (1) random splitting, (2) splitting based on K-means clustering (both

of actives and inactives), (3) splitting based on source database and (4) splitting based both in the clustering and in the source database. These schemas are applied to a Deep Learning proteochemometrics model and to a simple logistic regression model to be used as baseline. Additionally, two different ways of describing molecules in the model are tested: (1) by their SMILES and (2) by three fingerprints. The classification performance of our Deep Learning-based proteochemometrics model is comparable to the state of the art. Our results show that the lack of generalization of these models is due to a bias in public molecular databases and that a restrictive cross-validation schema based on compounds clustering leads to worse but more robust and credible results. Our results also show better performance when representing molecules by their fingerprints.

# Introduction

Proteochemometrics or quantitative multi-structure-property-relationship modeling (QMSPR) is an extension from the traditional quantitative structure-activity relationship (QSAR) modeling.[1] In QSAR, the target protein is fixed and its interaction with ligands (small molecules or compounds) is predicted only from ligands descriptors. On the contrary, the aim of proteochemometrics is to predict the binding affinity value by modeling the interaction of both proteins and ligands.[1] For this, a data matrix is built, each of its rows containing descriptors of both target and ligand linked to some experimentally measured biological activity. A statistical or machine learning method is then used to induce the model. The main advantages over QSAR are twofold: first, that the induced model can be applied for predictions of interaction with new proteins as well as ligands and second, that it can consider the underlying biological information carried by the protein as well as other possible cross-interactions of the ligand.

Deep learning (DL) is a branch of machine learning that stems from artificial neural networks, which are computational models inspired in the structure of the brain and the

interconnection between the neurons. DL is able to learn representations of raw data with multiple levels of abstraction.[2] These concepts started to be developed in the 1940s[3] but it was not until 2012 that there was a break through of the Deep Neural Networks (DNN).[4] Since then, DL has been successfully applied in natural language processing,[5] image recognition,[6] drug discovery[7] or computational biology.[8] The increase of computational power by parallel computing with graphics processing units (GPU) and the improvement of optimizers[9,10] and regularization techniques[11,12] contributed to this resurgence, along with the development of software platforms that allow to make prototyping faster and automatically manage GPU computing, like Theano[13] or Tensorflow.[14]

DL provides a framework for the identification of both of biological targets and biologically active compounds with desired pharmacological effects.[7] In 2012, DNN won a QSAR machine learning challenge on drug discovery and activity prediction launched by Merck,[15] outperforming Merck's Random Forests baseline model by 14% accuracy.[16] Since then, the application of DL to pharmaceutical problems gained popularity,[17–27] although it has been mainly applied to multitask QSAR modeling. Regarding DL-based proteochemometrics, little has been done except for the work of Lenselink et al,[28] where they compared different machine learning methods for proteochemometrics, being DNN the top performer.

Independently of the machine learning technique used, a curated design of the cross-validation strategy is critical for the proper evaluation of the binding prediction model. The predictive power of a consistent model must remain stable when applied to data that comes from a different source than the training set. Moreover, possible redundancy in the data must be controlled. Proteins are divided into families, which usually have similarities in sequence or structure. Compounds might be part of the same chemical series. The performance of classification model should be tested when applied to families of proteins or compounds with different scaffolds than those used to train it. On the latter, Wallach and Heifets concluded that performance of most of the reported ligand-based classification problems reflect overfitting to training benchmarks rather than good prospective accuracy,[29]

mainly because of the redundancy between training and validation sets. This issue becomes more critical when using random cross-validation: in the pharmaceutical field, compounds are usually synthesized serially to enhance molecular properties. This leads to training and validation sets following the same distribution, which is desirable in most machine learning problems, but a poor estimate of reality in drug discovery.[22]

Time-split validation is common practice in pharmaceutical environment to overcome this issue.[22,28,30,31] This strategy is well suited to the realistic scenario, where we are interested in prospective performance of the models.[30] However, most public data lack of temporal information, hindering this strategy to be applied. Additionally, time-split data has also shown to be biased because of the high similarity between discovered actives in different phases.[22,29]

Other techniques have been applied to reduce bias and data redundancy between training and validation sets. Unterthiner et al.[17,27] clustered compounds using single linkage to avoid having compounds sharing scaffolds across training and validation sets. Rohrer and Baumann designed the Maximally Unbiased Validation (MUV) benchmark to be challenging for standard virtual screening: actives have been selected to avoid biases of enrichment assessment and inactives have been biologically tested against their target.[32] Xia et al.[33] presented a method to ensure chemical diversity of ligands while keeping the physicochemical similarity between ligands and decoys. Wallach et al. removed analogue bias in active molecules by clustering and selected decoys to match in sets to actives with respect to some 1D physicochemical descriptors while being topologically dissimilar based on 2D fingerprints.[18] However, these unbiasing techniques only focus on redundancy between actives, overlooking the impact of inactive-active or inactive-inactive similarity, which leads to models memorizing the similarity between benchmark inactives and hence, overfitting.[29]

Another related issue is that the possible bias across the different data sources used in some studies has not been properly studied yet.[18] Different datasets might have different structure, affecting to the study of the generalization of the model. A related issue is found in

the study of Altae-Tran et al,[34] where after the collapse of their transfer learning experiments it is affirmed that one-shot learning methods may struggle to generalize to novel molecular scaffolds, and that there is a limit to their cross-tasks generalization capability.

Analysis of bias in binding classification models have been always focused on QSAR models, but how could affect the inclusion of proteins to bias in QMSPR models remains unknown. Proteins are macromolecules constituted by amino acid residues covalently attached to one another, forming long linear sequences which identify them, defining its folding and its activity. The main value of DL in this context is that DL can directly learn from the sequence, capturing nonlinear dependencies and interaction effects, and hence providing additional understanding about the structure of the biological data. The appropriate DL architecture to manage this kind of data are bi-directional Recurrent Neural Networks (RNN), well suited for modeling data with a sequential but non-causal structure, variable length and long-range dependencies.[35] Baldi et al have applied bi-directional RNN to protein sequence for predicting secondary structure,[36–38] for matching protein beta-sheet partners[39] or for predicting residue-residue contact.[40] However, classical RNN cannot hold very long-range dependencies and to overcome this issue Hochreiter et al applied Long Short Term Memory (LSTM) networks to classify amino acid sequences into superfamilies.[41] Jurtz et al applied bi-directional LSTM to amino acid sequence for subcellular localization, secondary structure prediction and peptides binding to a major histocompatibility complex.[42]

In this paper, we analyse and quantify the effect of different cross-validation strategies on the performance of binding prediction DL-based proteochemometrics models. Additionally, we compare these DL models with baseline logistic regression (LR) models and explore different representations for molecules.

Table 1: Number of targets, number of compounds, average number of compounds per target and average percentage of actives for each source database present in the Riniker et al dataset[43] after adapting it for our study. SD: standard deviation.

| Original database | # targets | # compounds | # compounds/target mean (SD) | % actives mean (SD) |
|---|---|---|---|---|
| ChEMBL | 50 | 29,986 | 599.7 (0.9) | 16.7 % (0.1) |
| DUD | 21 | 12,417 | 591.3 (80.5) | 14.2 % (9.4) |
| MUV | 17 | 9,001 | 529.5 (1.1) | 5.7 % (0.0) |
| **Total** | **88** | **51,404** | **584.1 (47.2)** | **13.9 % (6.1)** |
| **Unique** | 83 | 32,950 | - | - |

# Materials and Methods

## Data

Models were trained on the dataset generated from three different publicly available sources by Riniker and Landrum[43] for true reproducibility and comparability of benchmarking studies. This dataset incorporates 88 targets from ChEMBL,[44] the Directory of Useful Decoys (DUD)[45] and the MUV.[32] The selection of actives and decoys was conducted on drug-like molecules and in such a way as to cover the maximum range of the chemical spectrum, based on diversity and physical properties. ChEMBL and DUD decoys were selected from the ZINC database.[46] The selection of ChEMBL targets was based on the 50 human targets and actives proposed by Heikamp and Bajorath study[47] and performed on ChEMBL version 14.[43]

We only selected 500 decoys randomly from all those available for each target, in order to have a more computationally-approachable dataset and to decrease active/decoy imbalance per target while keeping a plausible proportion. The list of molecules identified by their SMILES[48] was then standardized to avoid multiple tautomeric forms. Finally, these compounds were filtered to remove salts, those with molecular weight >900Da or >32 rotatable bonds and those containing elements other than C, H, O, N, S, P or halides. Table 1 provides a summarized description of the final dataset used in our study, while Table S1 of the Supporting Information contains a more detailed description.

## Descriptors

We tested two ways of representing input molecules: (1) as sequences of symbols, using the SMILES notation and (2) as the combination of molecular fingerprints,[49] where structural information is represented by bits in a bit string. The SMILES representation as input for a DL model was based on the DeepCCI by Kwon et al.[26] Model input has to be numerical, so SMILES notation was one-hot encoded (Figure 1B). This means that every character of the SMILES string was represented by a binary vector of size 35, with all but its corresponding entry set to zero. SMILES were padded to the length of the longest string, 94.

For the fingerprints representation we selected three of them: topological torsions (TT) fingerprint,[50] extended connectivity fingeprint and functional connectivity fingerprint, both with a diameter of 6 (ECFP6 and FCFP6, respectively)[51] (Figure 1C). TT describe four atoms forming a torsion, and the atom type includes the element, the number of non-hydrogen neighbors and number of $\pi$ electrons. ECFP6 and FCFP6 encode circular atom environment up to 6 bond length. In ECFP6, atom type includes the element, the number of heavy-atom neighbors, the number of hydrogens, the isotope and ring information. FCFP6 use pharmacophoric features. All of them were generated using the RDKit package,[52] and defined with a length of 1024 bits, since there is proof of a very low number of collisions with this size.[43]

For protein representation, raw amino acid sequences were fed to the model (Figure 1A). As for SMILES strings, these sequences were converted to numerical through one-hot encoding, only that in this case each amino acid was represented by a binary vector of length 20. Amino acid sequences were then padded to the length of the longest target, in this case, 1988.

## Cross-validation strategies

Four different cross-validation strategies were applied to both active and inactive Riniker dataset compounds (see Figure 2A), omitting binding targets. In all cases active/inactive

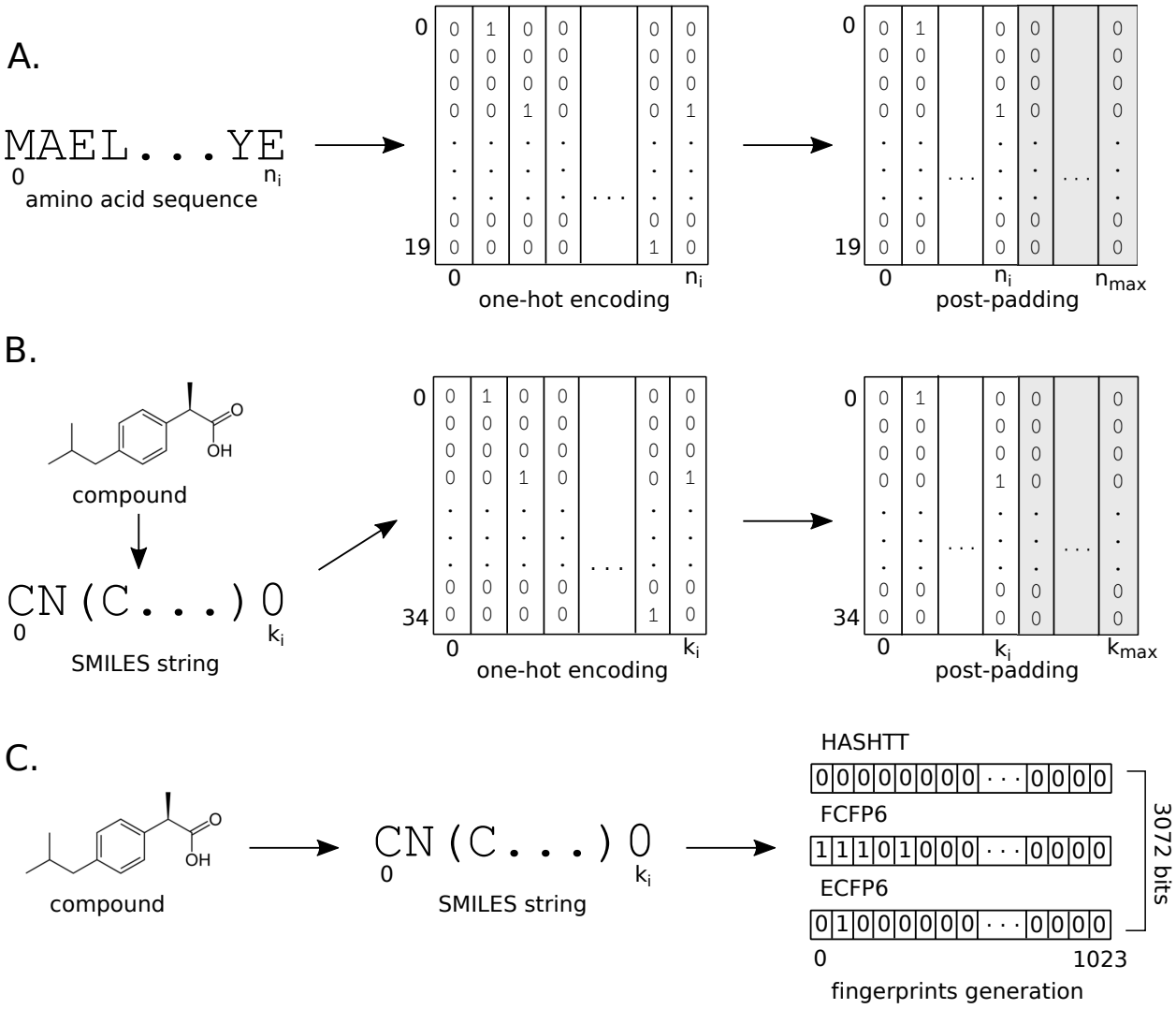


Figure 1: **Schema of descriptors and encoding process of the different inputs of the model A.** Amino acid sequence is first one-hot encoded and then padded at the end to the length of the longest target ($n_{max}$), in this case 1987. **B.** Compound represented by its SMILES identifier is also one-hot encoded and then padded to the length of the longest SMILES string ($k_{max}$), in this case 93. **C.** Compound described by its fingerprints is first identified by its SMILES, from which HASHTT, FCFP6 and ECFP6 bit strings are generated.

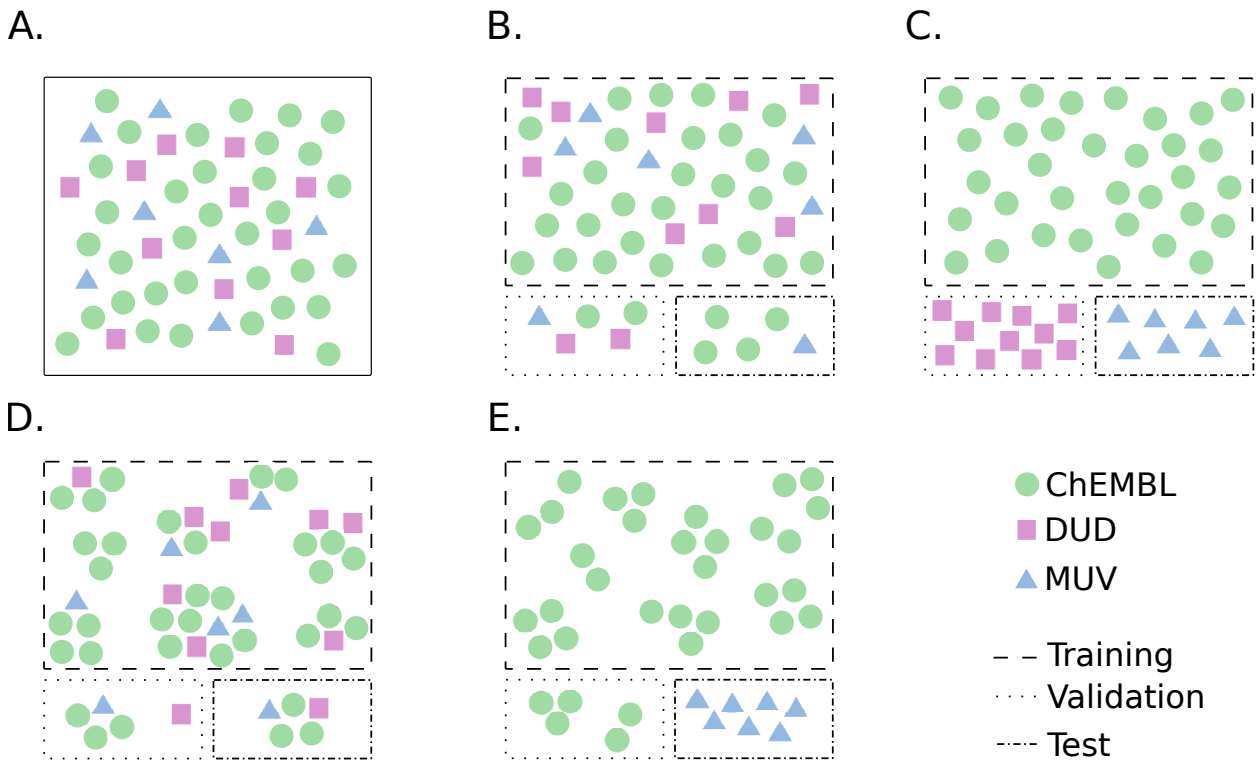


Figure 2: **Different cross-validation strategies** applied to the Riniker et al dataset.[43] **A.** Original dataset **B.** Random splitting of the compounds. **C.** Database-based division of compounds. **D.** Clustering-based splitting. **E.** Intermediate splitting.

proportion was preserved for training, validation and test sets. (1) **Random**, where compounds were randomly split 80/10/10 in training, validation and test with no further criteria (Figure 2B). (2) **Database-based**, where division in training, validation and test was performed according to the source database of the compounds (ChEMBL, MUV and DUD, as seen in Data section) (Figure 2C). (3) **Clustering-based**, where K-Means clustering with k=100 was applied to the fingerprint description of molecules (see Figure S1 of the Supporting Information). This was used to avoid having similar molecules both in training and validation/test set and thus control for the compound series bias.[27,29] Clusters were randomly joined and assigned to the splitting sets in order to have 80/10/10 splitting (Figure 2D). (4) **Intermediate**, where the previous K-Means clustering was also applied, but only to those compounds coming from ChEMBL (Figure 2D). In order for this schema to have a test set of comparable size with the others, only one data source was used. We chose MUV dataset since it was designed to be challenging, as seen in the Introduction, while data architectural design of the original DUD is not that well suited for this problem.[53] In Figures S2, S3, S4 and S5 of the Supporting Information, the proportion of actives/inactives for each target in each splitting set is depicted for random, clustering, database-based and intermediate cross-validation strategies, respectively.

## Prediction models

A DL-based model was built to evaluate the effect of different cross-validation strategies on binding prediction. A LR model was also trained to have a baseline to compare with. Besides, two different molecular representations were tested to evaluate their performance: their SMILES string and three selected fingerprints (ECFP6, FCFP6 and HASHTT). Each one of these models was trained according to the four cross-validation strategies presented in the previous section. As a result, a total of 16 different models were evaluated.

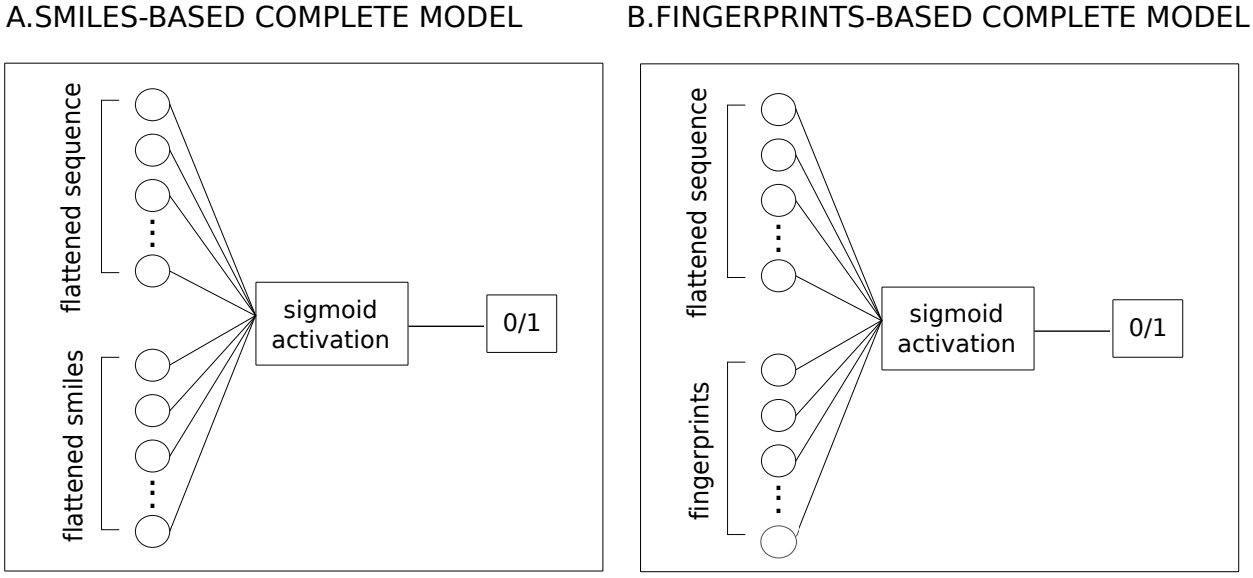


Figure 3: **Schematic representation of the baseline logistic regression model A.** When the compound is encoded by its SMILES, both amino acid sequence and SMILES inputs have to be flattened before entering the neurons of the input layer. **B.** When the compound is represented by its fingerprints, only the amino acid sequences have to be flattened.

### Logistic Regression

This baseline model consisted on two input layers concatenated, one for the compound and one for the target, with as many neurons as the size of the input (94 and 3072 in the case of SMILES and fingerprints, respectively and 1988 in the case of targets) connected to a sigmoidal unit (see Figure 3).

### Deep Learning models

An schematic representation of the DL predictive models used can be seen in Figure 4, A and B. The amino acid sequence analysis block is common for both models and it is a Convolutional Recurrent Neural Network based on the one used by Jurtz et al for the prediction of subcellular localization of proteins[42] (Figure 4C). This architecture allows to build complex representations from both targets and compounds for the prediction of binding, in contrast to the LR models, which are directly fed with the input features.

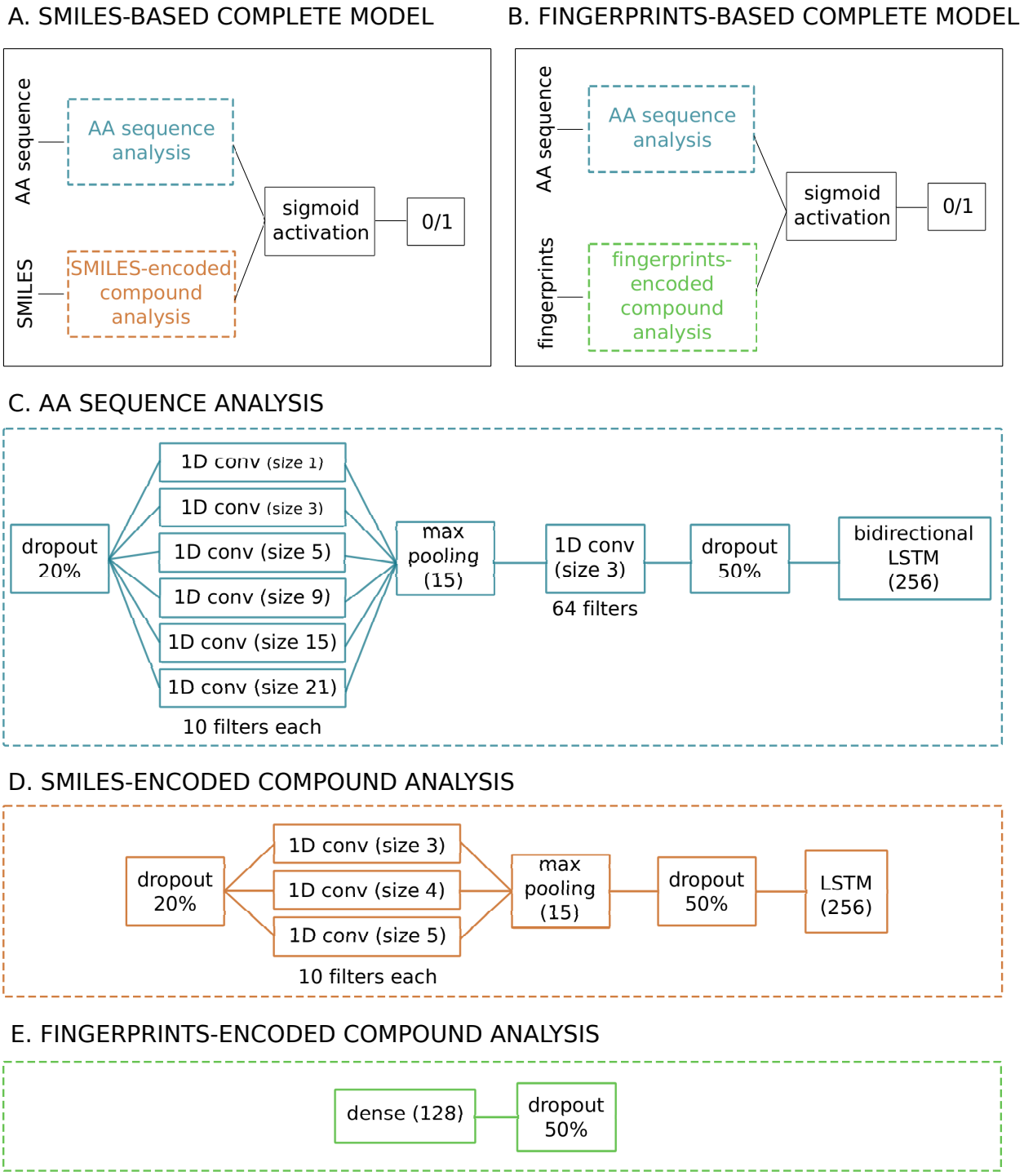


Figure 4: **Schematic representation of Deep Learning models.** AA: amino acids. (#) after the name of a layer refers to the number of neurons. **A.** Complete DL model when the compound is represented by their SMILES. **B.** Complete DL model when the compound is represented by their fingerprints. **C.** Amino acids sequence analysis block, a Convolutional Recurrent Neural Network architecture adapted from the model used by Jurtz et al.[42] **D.** SMILES-encoded compound analysis block., a Convolutional Recurrent Neural Network. **E.** Fingerprints-encoded compound analysis block, a feed-forward deep neural network.

The input of the model is the amino acid sequence one-hot encoded, which is passed through a 1D Convolutional Neural Network (CNN).[4] This 1D CNN comprises filters of sizes 1, 3, 5, 9, 15 and 21, with the aim of detecting motifs of different length in the amino acid sequence. Convolutional layers are followed then by a max pooling layer, downsampling the input and thus, reducing the number of model parameters. The input is then introduced to a bi-LSTM neural network.[35] Dropout algorithm is used in different parts of the model to prevent overfitting.[12]

The compound analysis block depends on the encoding of molecules. If molecules are represented by their SMILES, then the compound processing block is similar to the sequence processing block (Figure 4A). The SMILES string is one-hot encoded and the input is passed to a bank of convolutional filters, in this case of size 3,4 and 5 based on the sizes of the LINGO substrings analysed by Vidal et al.[54] After that, a maximum pooling layer condenses information and transfer it to a LSTM, but in this case uni-drectional since the SMILES strings are causal, in the sense that they are read in only one direction. Dropout is also used here to prevent overfitting. If molecules are represented by ECFP6, FCFP6 and HASHTT fingerprints, namely a binary vector of length 3072, the input is passed through a feed-forward neural network[17,19] followed by dropout (Figure 4E).

Finally, the sequence and the compound analysis blocks are merged and the information is processed by a sigmoid activation unit, which quantifies how likely the sequence-compound binding is. Binary predictions are obtained thresholding the activation at 0.5. All the models (both DL and LR) were trained with Adam optimizer[10] for 500 epochs, with a batch size of 128 for training and 64 for validation (learning rate=5e-6 for DL models encoded by fingerprints, 5e-5 for the rest). Decay rate was defined as learning rate/number of epochs. The other parameters were set as proposed by Jurtz et al ($\beta_1$=0.1, $\beta_2$=0.001, $\epsilon$=1e-8).[42]

## Implementation

Both the DL and the LR algorithms were implemented in Python (Keras[55] > 2.0 using as backend TensorFlow[14] > 1.4) and run on the GPU NVIDIA TITAN Xp and NVIDIA GeForce GTX 1070.

## Characterization of data structure

Each one of the cross-validation strategies is analyzed in terms of imbalance and data redundancy to better understand and interpret performance results. First, active/inactive proportion is explored for each cross-validation schema. Then, overlap of targets and compounds between split sets is computed as a percentage with respect to the total number of targets (88) and compounds (32,950), respectively. Lastly, distribution of chemotypes and protein classes is explored for each strategy. For targets, this distribution is studied for the main protein families. Since for molecules there is no such classification, we decided to group them in terms of their Bemis-Murcko scaffold (BMS),[56] a technique for extracting molecular frameworks by removing side chain atoms which has been used for clustering compounds.[18,24,53]

## Performance metrics

Area under the receiver operating characteristic (ROC) curve (from now, referred to as AUC), traditionally employed for measuring the performance of classification models, has been reported for not being enough for evaluating virtual screening models because it is not sensitive to early recognition and it is affected by class imbalance.[57] Thus, we complement this information with partial AUC (pAUC) at 5%, which allows to focus on the region of the ROC curve more relevant for virtual screening[1] (up to 5% of the False Discovery Rate), with Cohen's kappa coefficient ($\kappa$),[58] which measures the agreement between real and predicted classification, and with the Boltzmann-enhanced discrimination of receiver operating

characteristic (BEDROC), a metric proposed to overcome the limitations of AUC[57] increasingly popular in the evaluation of virtual screening models. BEDROC uses an exponential function based on parameter $\alpha$ and is bounded between 0 and 1, making it suitable for early recognition. As recommended by Riniker et al,[43] we focus on AUC and BEDROC with $\alpha$=20, whilst also reporting $\alpha$=100.

We implemented and trained DL and LR binding classification models. We then selected the best training epoch in terms of F1 Score, the harmonic mean of precision and recall, on the validation set, since it can handle class imbalance. Finally, we tested the selected models on the corresponding test set of each cross-validation strategy. Stratified subsampling of the 80% of the test data was used to sample 100 values from the performance estimates distributions. The nonparametric Wilcoxon rank-sum test[59] was used to compare AUC and BEDROC(20) metrics between all pairs of models. P-values were adjusted for multiple testing by computing the False Discovery Rate (FDR) by Benjamini-Hochberg[60] for each metric.

# Results

## Cross-validation strategies analysis

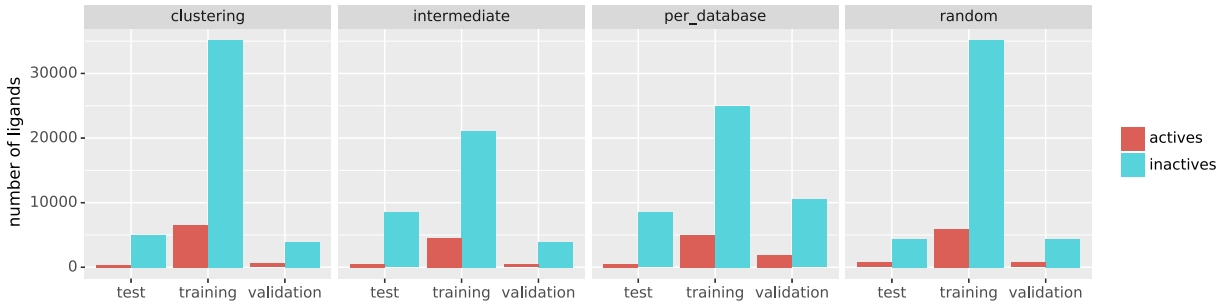


Figure 5: Proportion of active/inactive compounds in each set (training, validation and test), for each cross-validation strategy.

In Figure 5, active/inactive imbalance in each split set is shown for each cross-validation

strategy. Active/inactive proportion maintains for all sets.

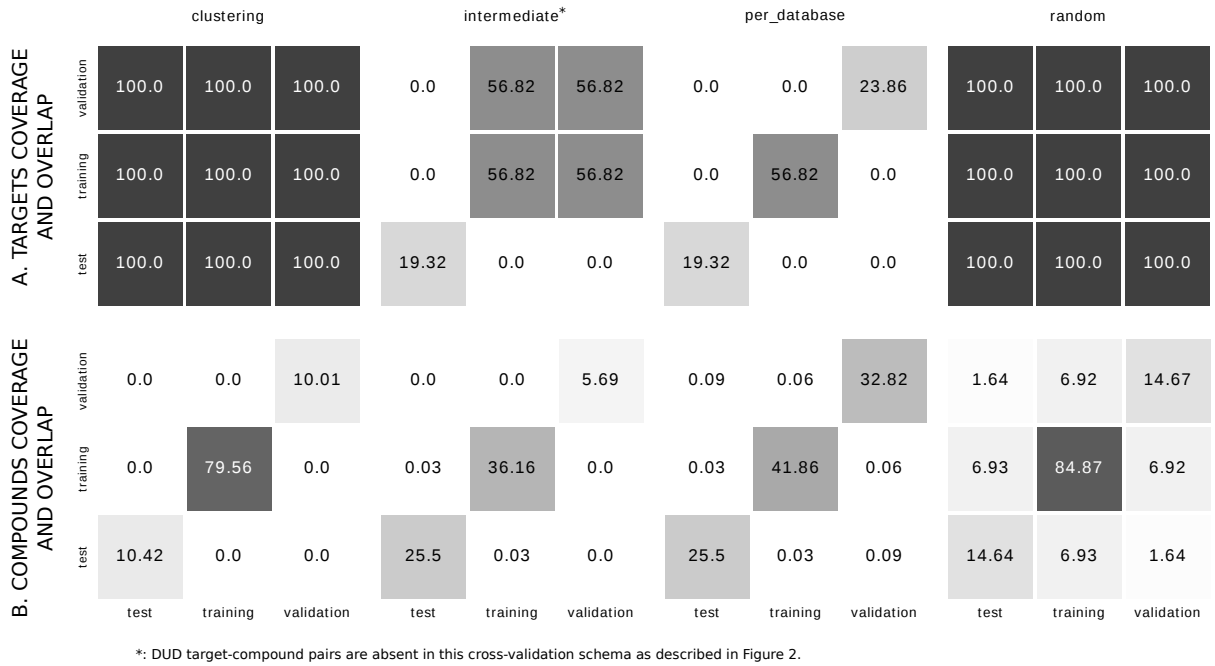


Figure 6: **A.** Coverage and overlap of targets between splitting sets. Numbers inside tiles refer to the percentage of overlapping targets respect to the total number of targets, 88. **B.** Coverage and overlap of compounds between splitting sets. Numbers inside tiles refer to the percentage of overlapping compounds respect to the total number of targets, 32,950.

Figure 6A shows overlap of targets between split sets for each cross-validation schema. In clustering-based and random, every target is repeated on the three splitting sets. In intermediate we only find overlapping targets between training and validation, since both splitting sets are built from the ChEMBL dataset, while test corresponds to the MUV database. In database-based, there is no overlap of targets between splitting sets.

Figure 6B shows overlap of compounds between splitting sets for each cross-validation schema. In this case, in clustering-based there are no repeated compounds across groups. In database-based and intermediate there is negligible overlap between groups, probably due to repeated inactives for different targets. Since in random cross-validation splitting was made randomly, there are repeated compounds in all splitting sets.

In Figure 7, Bemis-Murcko molecular scaffolds overlap between split sets is shown. As

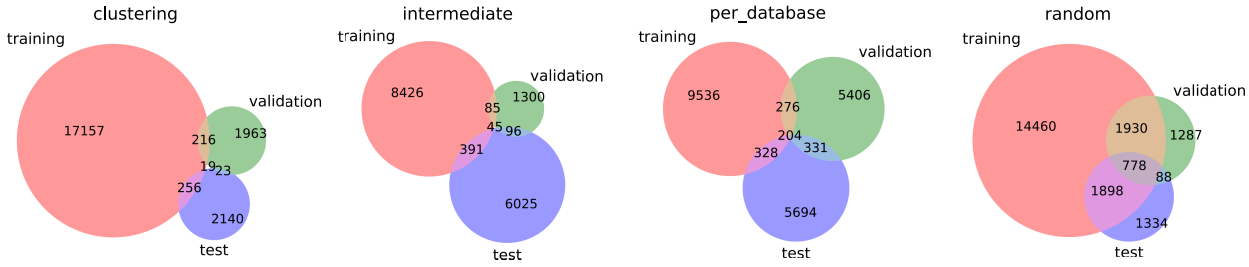


Figure 7: Overlap of BMS between split sets for each cross-validation strategy.

above, in random cross-validation there are more overlapping scaffolds between training, test and validation. In the other strategies the number of overlapping scaffolds decreases significantly. Training set in clustering-based cross-validation has the highest number of different scaffolds.

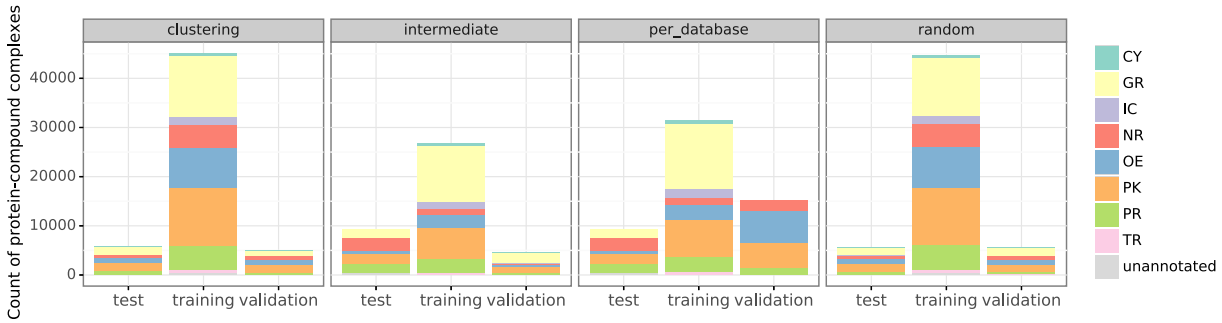


Figure 8: Distribution of protein families between split sets for each cross-validation strategy. CY: cytochromes P450, GR: G protein-coupled receptors, IC: ion channels, NR: nuclear receptors, OE: other enzymes, PK: protein kinases, PR: proteases, TR: transporters.

In Figure 8 distribution of protein families between split sets for each validation strategy is represented. G protein coupled-receptors and protein kinases are the most numerous families in training sets. Every group has targets of each protein family, except for the validation set in the database-based schema which lacks some protein families (transporters, cytochromes, G protein-coupled receptors and ion channels).

## Models performance

A summary of performance metrics of each model can be seen on Table 2.

Table 2: Summary of performance of different cross-validation strategies for DL and LR models, in terms of F1 score, AUC, partial AUC (pAUC) at 5%, BEDROC($\alpha$=20) and BEDROC($\alpha$=100). In bold, the best performance for each CV strategy. CV: cross-validation.

| CV strategy | Encoding | Algorithm | Best epoch | F1 score | AUC | pAUC(5%) | BEDROC(20) | BEDROC(100) | $\kappa$ |
|---|---|---|---|---|---|---|---|---|---|
| Random | Fingerprints | LR | 496 | 0.71 | **0.91** | 0.029 | 0.87 | 0.95 | 0.677 |
| | | DL | 348 | **0.88** | 0.89 | **0.031** | **0.89** | **0.99** | **0.838** |
| | SMILES | LR | 499 | 0.54 | 0.81 | 0.018 | 0.74 | 0.97 | 0.465 |
| | | DL | 432 | 0.47 | 0.75 | 0.016 | 0.68 | 0.94 | 0.436 |
| Clustering | Fingerprints | LR | 491 | 0.36 | **0.84** | 0.015 | **0.53** | 0.55 | 0.360 |
| | | DL | 452 | **0.68** | 0.74 | **0.017** | **0.53** | **0.72** | **0.445** |
| | SMILES | LR | 489 | 0.29 | 0.73 | 0.012 | 0.42 | 0.52 | 0.278 |
| | | DL | 463 | 0.25 | 0.66 | 0.008 | 0.31 | 0.45 | 0.250 |
| Database based | Fingerprints | LR | 22 | 0.25 | 0.54 | 0.001 | 0.10 | 0.04 | 9.47e-4 |
| | | DL | 19 | **0.51** | 0.54 | **0.002** | **0.11** | **0.09** | **0.043** |
| | SMILES | LR | 230 | 0.08 | 0.50 | 0.001 | 0.08 | 0.06 | 7.16e-4 |
| | | DL | 494 | 0.14 | **0.55** | **0.002** | 0.10 | **0.09** | 0.031 |
| Intermediate | Fingerprints | LR | 57 | 0.60 | 0.52 | 0.001 | 0.09 | 0.05 | -0.002 |
| | | DL | 309 | **0.62** | 0.54 | 0.002 | 0.13 | 0.11 | 0.032 |
| | SMILES | LR | 428 | 0.40 | 0.50 | 0.001 | 0.08 | 0.06 | -2.77e-4 |
| | | DL | 493 | 0.38 | **0.55** | **0.003** | **0.14** | **0.13** | **0.034** |

Models based on random cross-validation schema had consistently the best performance for both DL and baseline LR algorithms. In all cases, except in the intermediate cross-validation DL model, the architecture based on the fingerprints representation of compounds had a better performance than the SMILES encoding. Regarding the algorithm used, results are not conclusive on which one performs better. In terms of AUC, the best performance is from a logistic regression model and in terms of BEDROC, a deep learning model. In general, the best algorithm depends of the cross-validation strategy and the compound representation, but there is a tendency of logistic regression being better for the SMILES representation of the compound, and deep learning for the fingerprints representation.

In Figure 9, performance of the different cross-validation strategies is better depicted: the values of AUC and BEDROC(20) for each schema, compound representation and algorithm are compared between them and with a random prediction. Error bars show the standard deviations obtained from subsampled estimates. It can be seen that random strategy outperforms the rest in all the possibilities. After random, clustering-based strategy had the best performance both in terms of AUC and BEDROC(20). Database-based and intermediate strategies have both poor performance, specially in terms of BEDROC(20). The same behavior can be seen in ROC curves of all possibilities in Figure S6 of the Supporting

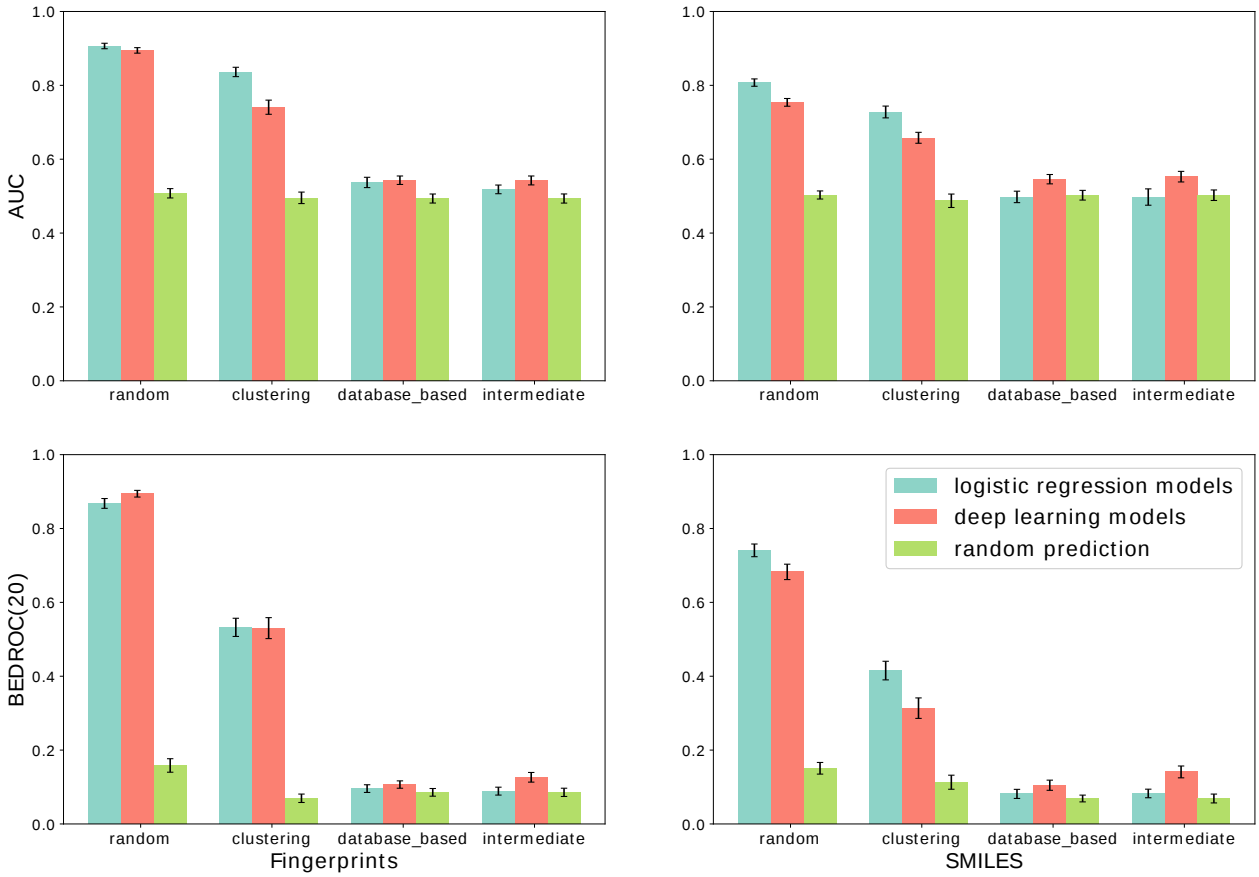


Figure 9: Comparison of performance of the different models, grouped by cross-validation strategies and colored by algorithm used for the prediction. Top row compares results in terms of AUC and bottom row in terms of BEDROC ($\alpha = 20$) score. In blue, logistic regression models metrics are shown; in red, deep learning-based models, and in green, a set of random prediction synthetically generated. Left column shows results for fingerprints-encoded models and right column for SMILES-encoded models. Error bars indicate standard error of the mean.

Information.

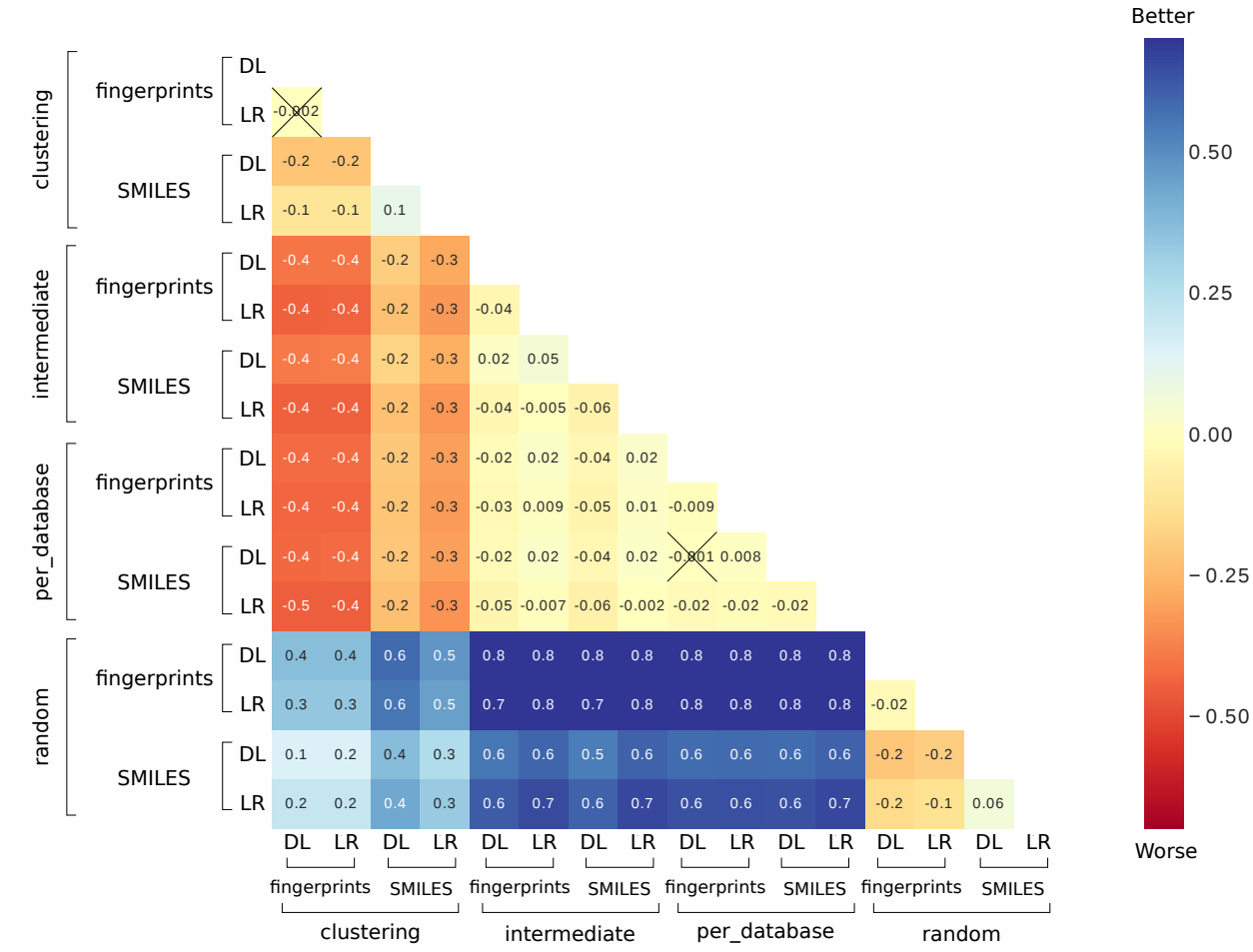


Figure 10: Mean difference for BEDROC(20) metrics between pairs of models. Differences are calculated subtracting column performances from rows. Crossed tiles indicate that difference for that pair of models is not statistically significant ($\alpha = 0.05$).

The mean difference of BEDROC(20) metrics for each possible combination of models is shown in 10. Differences in terms of AUROC follow the same behavior but of a smaller magnitude (see Figure S7 of Supplementary Information). The most remarkable differences can be seen on random models versus database-based and intermediate models. There are also relevant differences on performance between clustering-based cross-validation and the rest of strategies.

# Discussion

## Cross-validation strategy

The choice of cross-validation strategy is the most influential factor in this study. In general terms, random cross-validation outperforms the other schemes. When adding constraints for set splitting, performance suffers a pronounced drop. Such disagreements between cross-validation strategies suggest that compounds from different databases have different properties, i.e. that molecular databases may be biased. This bias would explain why models struggle to generalize between databases. On the other hand, any limitations derived from the data selection and benchmark construction by Riniker and Landrum might affect the predictive power of the models.

Our results show that random cross-validation leads to the best performance estimates for all the models. Likewise, random cross-validation shares the most proportion of compounds and proteins between the training, validation and test folds (Figures 6A and 6B). This is also true at the molecular scaffolds level (Figure 7), suggesting that shared scaffold inflate the performance estimates. Despite this, random cross-validation has been traditionally used in literature to evaluate binding prediction models. Our results are in line with previous reports suggesting that the reported accuracy of most published virtual screening models is unreliable.[22,29]

Among the non-random cross-validation strategies, database-based and intermediate schemes lead to models barely outperforming a random predictor in terms of AUC and BEDROC(20), see Figure 9 and Table 2. These schemes are probably too conservative, because the protein families and their proportions in their training, validation and test folds differ (Figure 8). In addition, their number of scaffolds in train (respectively 10,344 and 8,947) is smaller than random (19,066) and clustering (17,648) strategies, see Figure 7, limiting the variety of examples the models can learn from the training data. The clustering strategy allows training with more scaffolds and keeping a similar proportion of protein

families, while controlling for the data redundancy issue as the number of scaffolds shared between train and test decreases from 2,676 (random) to 275. It also leads to less optimistic performance estimates than the random strategy, whereas the models still retain predictive power. Therefore, the clustering strategy is chosen as our reference, in line with previous studies.[27]

## Compound encoding

In general, models for compounds represented by their fingerprints outperformed models for SMILES representation in clustering-based and random strategies, for both AUC and BEDROC(20), see Figure 9. This can be due to the specific architectures employed in each case: the compound analysis block for SMILES-encoding is based on CNN and LSTM to capture the sequence structure, while the compound analysis block for fingerprints is based on a single feed-forward neural network. This difference in model complexity and by extension, in the number of parameters, could have resulted in a poorly fitted SMILES-encoded model. Given that fingerprints-based models show a better performance, we will focus on them from this point on.

## Prediction algorithms

Regarding the algorithm used within the clustering strategy and FP encoding, DL and LR appear technically tied at a BEDROC(20) around 0.53 (Table 2). LR outperforms DL in terms of AUC (0.84 versus 0.74), but falls behind in the early recognition metrics pAUC(5%) (0.015 versus 0.017), BEDROC(100) (0.55 versus 0.72) and other metrics such as Cohen's Kappa (0.360 versus 0.445) and F1 score (0.36 versus 0.68). In agreement with previous studies,[57] the AUC appears misleading for early recognition. Despite the tie between LR and DL, the alternative metrics favor DL and is therefore preferred over LR by a small margin.

On the other hand, performance of the random cross-validation fingerprints-based DL

model (BEDROC(20) of 0.89) is slightly lower to the other published DL-based proteochemometrics model[28] (0.96). Lenselink et al represent proteins through standard physicochemical descriptors, whereas we use their amino acid sequence. The fact that amino acid-based representations attain a good predictive power poses the opportunity to gain insights into the mechanisms causing protein-ligand binding by analyzing biological patterns in the CNN filters and the long-range dependencies in the LSTM. Regarding the validation, Lenselink et al apply a temporal split strategy that drops the BEDROC metric 0.11 units, while our clustering strategy penalizes 0.33 units to our DL model. This is expected as time-split cross-validation can still suffer from chemical series bias.[22,29]

# Conclusion

We have benchmarked protein-compound binding models using two molecular representations for compounds and two prediction algorithms under four cross-validation strategies. One of our main findings is the existence of a database-specific bias that challenges the generalization of machine learning models between databases. Performance estimates derived from classical random cross-validation are overly optimistic, despite being widely used in literature. Instead, we recommend a clustering-based cross-validation since it addresses the chemical series bias while providing more reliable performance estimates. For molecular representation, fingerprints have led to better models than the SMILES identification string. Regarding the prediction algorithm, deep learning models have a small edge over a baseline logistic regression and will allow further interpretation of the CNN-LSTM architecture to provide valuable information on the binding mechanisms.

# References

(1) Andersson, C. R.; Gustafsson, M. G.; Strömbergsson, H. Quantitative Chemogenomics: Machine-Learning Models of Protein-Ligand Interaction. *Curr. Top. Med. Chem.* **2011**,

*11*, 1978–1993.

(2) LeCun, Y.; Yoshua, B.; Geoffrey, H. Deep learning. *Nature* **2015**, *521*, 436–444.

(3) McCulloch, W.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133.

(4) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst. 25* **2012**, 1097–1105.

(5) Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent Trends in Deep Learning Based Natural Language Processing. *arXiv.org* **2018**, No. 1708.02709.

(6) He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. 2016 Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2016; pp 770–778.

(7) Gawehn, E.; Hiss, J. A.; Schneider, G. Deep Learning in Drug Discovery. *Mol. Inf.* **2016**, *35*, 3–14.

(8) Deep learning for computational biology. *Mol. Syst. Biol.* **2016**, *12*.

(9) Hinton, G.; Srivastava, N.; Swersky, K. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. 2012.

(10) Kingma, D. P.; Ba, J. L. Adam: A Method For Stochastic Optimization. *arXiv.org* **2014**, No. 1412.6980.

(11) Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv.org* **2015**, 448–456.

(12) Srivastava, N.; , G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.

(13) Theano Development Team, Theano: A Python framework for fast computation of mathematical expressions. *arXiv.org* **2016**, No. 1605.02688.

(14) Abadi, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015; `http://tensorflow.org/`.

(15) Merck Activity Competition. 2012; `https://www.kaggle.com/c/MerckActivity`.

(16) Dahl, G. E.; Jaitly, N.; Salakhutdinov, R. Multi-task Neural Networks for QSAR Predictions. *arXiv.org* **2014**, No. 1406.1231.

(17) Unterthiner, T.; Mayr, A.; Klambauer, G.; Hochreiter, S. Deep Learning for Drug Target Prediction. NIPS Workshop on Deep Learning and Representation Learning. 2014.

(18) Wallach, I.; Dzamba, M.; Heifets, A. AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery. *arXiv.org* **2015**, No. 1510.02855.

(19) Ramsundar, B.; Kearnes, S.; Edu, K.; Riley, P.; Webster, D.; Konerding, D.; Pande, V. Massively Multitask Networks for Drug Discovery. *arXiv.org* **2015**, No. 1502.02072.

(20) Mayr, A.; Klambauer, G.; Unterthiner, T.; Hochreiter, S. DeepTox: Toxicity Prediction using Deep Learning. *Front. Environ. Sci.* **2016**, *3*, 80.

(21) Kadurin, A. et al. The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget* **2017**, *8*, 10883–10890.

(22) Kearnes, S.; Goldman, B.; Pande, V. Modeling Industrial ADMET Data with Multitask Networks. *arXiv.org* **2017**, No. 1606.08793.

(23) Tian, K.; Shao, M.; Wang, Y.; Guan, J.; Zhou, S. Boosting compound-protein interaction prediction by deep learning. *Methods* **2016**, *110*, 64–72.

(24) Gomes, J.; Ramsundar, B.; Feinberg, E. N.; Pande, V. S. Atomic Convolutional Networks for Predicting Protein-Ligand Binding Affinity. *arXiv.org* **2017**, No. 1703.10603.

(25) Neil, D.; Segler, M.; Guasch, L.; Ahmed, M.; Plumbley, D.; Sellwood, M.; Brown, N. Exploring Deep Recurrent Models with Reinforcement Learning for Molecule Design. *Workshop track - ICLR 2018* **2018**, 1–15.

(26) Kwon, S.; Yoon, S. DeepCCI: End-to-end Deep Learning for Chemical-Chemical Interaction Prediction. *arXiv.org* **2017**, No. 1704.08432.

(27) Mayr, A.; Klambauer, G.; Unterthiner, T.; Steijaert, M.; Wegner, J. K.; Ceulemans, H.; Clevert, D.-A.; Hochreiter, S. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chem. Sci.* **2018**,

(28) Lenselink, E. B.; Ten Dijke, N.; Bongers, B.; Papadatos, G.; van Vlijmen, H. W. T.; Kowalczyk, W.; IJzerman, A. P.; van Westen, G. J. P. Beyond the hype: deep neural networks outperform established methods using a ChEMBL bioactivity benchmark set. *J. Cheminf.* **2017**, *9*, 45.

(29) Wallach, I.; Heifets, A. Most Ligand-Based Classification Benchmarks Reward Memorization Rather than Generalization. *J. Chem. Inf. Model* **2018**, *58*, 916–932.

(30) Sheridan, R. P. Time-Split Cross-Validation as a Method for Estimating the Goodness of Prospective Prediction. *J. Chem. Inf. Model* **2013**, *53*, 783–790.

(31) Ma, J.; Sheridan, R. P.; Liaw, A.; Dahl, G. E.; Svetnik, V. Deep Neural Nets as a Method for Quantitative Structure–Activity Relationships. *J. Chem. Inf. Model.* **2015**, *55*, 263–274.

(32) Rohrer, S. G.; Baumann, K. Maximum Unbiased Validation (MUV) Data Sets for Virtual Screening Based on PubChem Bioactivity Data. *J. Chem. Inf. Model.* **2009**, *49*, 169–184.

(33) Xia, J.; Jin, H.; Liu, Z.; Zhang, L.; Wang, X. S. An Unbiased Method To Build Benchmarking Sets for Ligand-Based Virtual Screening and its Application To GPCRs. *J. Chem. Inf. Model.* **2014**, *54*, 1433–1450.

(34) Altae-Tran, H.; Ramsundar, B.; Pappu, A. S.; Pande, V. Low Data Drug Discovery with One-Shot Learning. *ACS Central Sci.* **2017**, *3*, 283–293.

(35) Lipton, Z. C.; Berkowitz, J.; Elkan, C. A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv.org* **2015**, No. 1506.00019.

(36) Baldi, P.; Brunak, S.; Frasconi, P.; Soda, G.; Pollastri, G. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics* **1999**, *15*, 937–46.

(37) Baldi, P.; Pollastri, G. The Principled Design of Large-Scale Recursive Neural Network Architectures–DAG-RNNs and the Protein Structure Prediction Problem. *J. Mach. Learn. Res.* **2003**, *4*, 575–602.

(38) Agathocleous, M.; Christodoulou, G.; Promponas, V.; Christodoulou, C.; Vassiliades, V.; Antoniou, A. Protein Secondary Structure Prediction with Bidirectional Recurrent Neural Nets: Can Weight Updating for Each Residue Enhance Performance? IFIP International Conference on Artificial Intelligence Applications and Innovations. 2010; pp 128–137.

(39) Baldi, P.; Pollastri, G.; Andersen, C. A.; Brunak, S. Matching protein beta-sheet partners by feedforward and recurrent neural networks. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **2000**, *8*, 25–36.

(40) Di Lena, P.; Nagata, K.; Baldi, P. Deep architectures for protein contact map prediction. *Bioinformatics* **2012**, *28*, 2449–2457.

(41) Hochreiter, S.; Heusel, M.; Obermayer, K. Fast model-based protein homology detection without alignment. *Bioinformatics* **2007**, *23*, 1728–1736.

(42) Jurtz, V. I.; Johansen, A. R.; Nielsen, M.; Almagro Armenteros, J. J.; Nielsen, H.; Sønderby, C. K.; Winther, O.; Sønderby, S. K. An introduction to deep learning on biological sequence data: examples and solutions. *Bioinformatics* **2017**, *33*, 3685–3690.

(43) Riniker, S.; Landrum, G. A. Open-source platform to benchmark fingerprints for ligand-based virtual screening. *J. Cheminf.* **2013**, *5*, 26.

(44) Bento, A. P.; Gaulton, A.; Hersey, A.; Bellis, L. J.; Chambers, J.; Davies, M.; Krüger, F. A.; Light, Y.; Mak, L.; McGlinchey, S.; Nowotka, M.; Papadatos, G.; Santos, R.; Overington, J. P. The ChEMBL bioactivity database: an update. *Nucleic Acids Res.* **2014**, *42*, D1083–D1090.

(45) Huang, N.; Shoichet, B. K.; Irwin, J. J. Benchmarking Sets for Molecular Docking. *J. Med. Chem.* **2006**, *49*, 6789–6801.

(46) Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. ZINC: A Free Tool to Discover Chemistry for Biology. *J. Chem. Inf. Model.* **2012**, *52*, 1757–1768.

(47) Heikamp, K.; Bajorath, J. Large-Scale Similarity Search Profiling of ChEMBL Compound Data Sets. *J. Chem. Inf. Model.* **2011**, *51*, 1831–1839.

(48) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Model* **1988**, *28*, 31–36.

(49) *Handbook of molecular descriptors*; Wiley-VCH, 2000; p 667.

(50) Nilakantan, R.; Bauman, N.; Dixon, J. S.; Venkataraghavan, R. Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors. *J. Chem. Inf. Model* **1987**, *27*, 82–85.

(51) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model* **2010**, *50*, 742–754.

(52) Landrum, G. RDKit: Open-source cheminformatics. `http://www.rdkit.org`.

(53) Mysinger, M. M.; Carchia, M.; Irwin, J. J.; Shoichet, B. K. Directory of Useful Decoys, Enhanced (DUD-E): Better ligands and Decoys for Better Benchmarking. *J. Med. Chem* **2012**, *55*, 6582–6594.

(54) Vidal, D.; Thormann, M.; Pons, M. LINGO, an Efficient Holographic Text Based Method to Calculate Biophysical Properties and Intermolecular Similarities. *J. Chem. Inf. Model* **2005**, *45*, 386–393.

(55) Chollet, F. Keras. `https://keras.io`, 2015.

(56) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem* **1996**, *39*, 2887–2893.

(57) Truchon, J.-F.; Bayly, C. I.; Truchon, J.-F. Evaluating Virtual Screening Methods: Good and Bad Metrics for the " Early Recognition " Problem. *J. Chem. Inf. Model* **2007**, *47*, 488–508.

(58) Cohen, J. A Coefficient of Agreement for Nominal Scales. *Educ. Psychol. Meas.* **1960**, *20*, 37–46.

(59) Hollander, M.; Wolfe, D. A. Nonparametric statistical methods. **1999**,

(60) Benjamini, Y.; Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *J. R. Stat. Soc. Series B Stat. Methodol.* **1995**, 289–300.

# Acknowledgement

The authors thank the NVIDIA Corporation for the donation of the Titan Xp GPU used to perform the analysis of this article. This research was partially supported by an Industrial

Doctorate grant from the Generalitat of Catalonia to A.L.-d.R. (DI 2016-080). This work was also supported in part within the framework of the Ministerio de Economía, Industria y Competitividad (MINECO) Grant TEC2014–60337–R), and the Centro de Investigación Biomédica en Red (CIBER) of Bioengineering, Biomaterials and Nanomedicine, an initiative of the Instituto de Salud“ Carlos III” (ISCIII).

## Supporting Information Available

Supporting Information Available: Detailed information on the benchmark dataset, figures on structure of data (Principal Component Analysis, proportion of actives per target for split set), ROC curves of the resulting models, mean differences for AUC metrics between models.